免憑證代理簽名及其盲簽名之擴張

Certificateless Proxy Signature and Its Extension to Blind Signature

左瑞麟
National Chengchi University, Taiwan, ROC
Email:
raylin@cs.nccu.edu.tw

陳力瑋
National Chengchi University, Taiwan, ROC
Email:
g9726@cs.nccu.edu.tw

陳淵順
National Chengchi University, Taiwan, ROC
Email:
g9736@cs.nccu.edu.tw

詹省三
National Chengchi University, Taiwan, ROC
Email:
g9718@cs.nccu.edu.tw

一、 摘要

在傳統的公開金鑰簽章系統中，用戶的公鑰需要一個可信第三方(Trusted Third Party-TTP)發給憑證來保證其可靠性。其後 Shamir 提出基於使用者身分的簽名機制(ID-based Signature) 儘管不需要憑證，但此種系統的概念中，TTP 仍然扮演著強大的角色，隨之而來的是 key escrow 的問題。而在 2003 年時提出的免憑證簽章系統 certificateless signature scheme(CL-S)概念中，不僅不需要憑證也同時解決了 key escrow 的問題。本篇文章便是基於 CL-S 的概念下，發展出一套免憑證的可代理簽章系統(CL-proxy signature)。並且可利用簡單的方式使我們的系統擴張成為一個支援盲簽名(Blind signature)的免憑證代理盲簽章系統。

In traditional public key infrastructure (PKI), a trusted third party called certificate authority (CA) is required in order to assure the correctness of a user's public key. ID-based cryptosystem solves this problem since a user's public key is just its identity. However, it suffers the key escrow problem. In 2003, Al-Riyami and Paterson introduced the certificateless cryptosystem which inherits both the advantages of traditional cryptosystems and ID-based cryptosystems. That is, a certificateless cryptosystem doesn't require a certificate and it can solve the key escrow problem. In this paper, we introduce a certificateless proxy signature scheme. Proxy signature is useful in many applications. The advantage of our scheme is that it is very easy and very efficient to extend our scheme into a certificateless proxy blind signature scheme. We emphasize that this is the first work that successfully constructing a certificateless proxy signature scheme.

關鍵詞： 免憑證簽章(Certificateless Signature)、雙線性配對(Bilinear Paring)、代理簽章(Proxy Signature)、盲簽章(Blind Signature)

二、緒論

在傳統的公開金鑰基礎架構(Public Key Infrastructure-PKI)中，公開金鑰簽章系統中需要有一個可信任的第三方(Trusted Third Party-TTP)

來發給使用者公鑰的憑證，認證者得到對方公鑰的時候，其中必有附一個關於公鑰的憑證，藉此憑證才可以使認證者確認此公鑰是真正的簽屬者發給的。PKI 是由管理電子憑證的許多機構所組成。其中最重要的組成元素即為驗證中心(Certificate Authority-CA)。CA 的主要工作就是簽發金鑰憑證，以提供系統使用者能取得所需他人的認證資料。此外，CA 亦需維護憑證資料庫以及定期發佈憑證註銷清單(Certification Revocation List)。PKI 在實作及運用上衍生出了許多問題，如金鑰憑證數量易於過度激增及憑證註銷清單過大等問題。這些問題不只降低 PKI 運作的效率，也增加了管理的成本。此外，在現今越來越要求頻寬的環境下，勢必是需要更省傳送空間的方式。

在 1984 年時，Shamir 提出了基於身分的數位簽章系統(ID-based signature scheme)[9]，大大減少了傳統公開金鑰系統中金鑰管理上的麻煩。在基於身分的簽章系統，使用者的公開金鑰部分是依據使用者本身獨一無二的資訊來產生個別的公開金鑰。這些資訊諸如：電子郵件帳號、身分證字號...等等。而在私鑰的部分，是經由可信任的第三方扮演一個 PKG(Private Key generator)的角色來產生與公開金鑰相配對的私鑰。在此系統架構中的使用者，可以減少在傳統公鑰系統中，使用者公鑰需要認證的手續。因此在 Shamir 提出這概念之後，便大量的發展出了許多相關的研究[3][4][8]，及其他方面的應用，例如電子投票、電子現金一類的運用。ID-based 系統排除了對金鑰憑證的需要和依賴，在一定程度上解決了現行 PKI 所遇到的問題。但是這種基於身分認證的系統架構中，可能會遇到金鑰託管(key escrow)的問題：PKG 知道所有使用者的公、私鑰。若是遇到不誠實的 PKG，便有可能遭受到偽造的攻擊。

其後，Al-Riyami 和 Paterson 在 2003 年發表了一種新的簽章概念：免憑證簽章系統(Certificateless Signature Scheme)[1]。這種簽章系統的主要概念便是希望除去傳統公開金鑰系統中的認證過程，以及基於身分認證系統中的 Key escrow 問題。在 CL-S 中同樣必須有個公正的第三方，在這裡稱為 KGC(key generator center)。KGC 並不會提供使用者一個完整的私鑰，取而代之的是利用使用者 ID 產生一個部分私鑰，使用者接受這個部分私鑰後，可選取亂數來和部分私鑰結合，進而產生只有自己知道的一組私鑰，並且使用者也會利用此亂數結合 KGC 所發佈的系統參數來產生一相對應的公鑰。結合以上幾種方法，便可以達成擁有傳統公開金鑰簽章以及基於身分認證簽章系統的優點。免憑證密碼系統因為不需驗證金鑰憑證，節省了許多複雜的計算，預計將可廣泛的應用於計算能力有限的電子機器像是手機或是 PDA 上。

另一方面，代理簽章法(Proxy signature)的概念是在 1996 年由日本的學者 M.Mambo 提出[7]，在代理簽章的系統下，系統中的原簽章者可將簽章的動作賦予一代理簽章者執行。且在 Mambo 的方法中，只有代理簽章者可以產生合法的代理簽章，並且使用者可以輕易地分辨簽章是由原簽章者亦或是代理者簽出，這個概念可以保障使用者一種公平性的保護。在 Mambo 的代理簽章法概念提出後，得到相當多的迴響以及研究[5][6][10]，在現今很多重要的應用都有使用到代理簽章的系統。

至於盲簽章(Blind Signature)的概念首先在 1982 年時，由 D.Chaumn 所提出[2]。盲簽章是個共同作用的簽章協定，文件的簽章是由使用者以及簽章者合作產生，在此系統中簽章者無處得知使用者欲簽名文件的內容，如此可以有效的保護使用者需要簽章的文件。

本篇文章便是基於以上幾種簽章模式，融合其優點來發展出在免憑證簽章系統下，原簽章者可以指定代理簽章者來完成簽章的工作。並且若使用者需要，還可以使此系統完成盲簽章的工

作，目前已知的方式大多只討論免憑證代理簽章的部分。B.Zhang 及 Q.Xu[11]雖於 2009 提出了一個免憑證代理盲簽章的方案，但經過檢驗，發現其演算法有嚴重的錯誤，所以，目前仍未有人提出此種簽章方式可包含盲簽名的部分，我們便往這簽章方式去做討論與設計。

文章剩下的部分，我們會先介紹一些背景知識以及此種系統的安全性要求，接著便會介紹我們所提出的免憑證代理簽章方案。再下一個部分，我們更將此方案改進成可運作於盲簽章的系統。其後會分析此套系統的安全性，並且在最後的部分做出結論。

二、背景知識

A. 雙線性配對 (Bilinear Pairing)

由於本篇文章中運用到雙線性配對的計算，故這裡先簡單的介紹相關的基本運算。

G_1為一加法群(Additive Group)，序(Order)為 q。

G_2為一乘法群(Multiplicative Group)，序也為 q。

P 是G_1的生成數(Generator)，則一個雙線性配對表示為$e: G_1 \times G_1 \rightarrow G_2$。

具有下列三種性質：

(1) 雙線性(Bilinear)：P, Q, R$\in G_1$及$a, b \in Z_q^*$，$e(aP, bQ) = e(P, Q)^{ab}$。

(2) 非退化性(Non-degenerate)：P, Q$\in G_1$，滿足$e(P,Q) \neq 1$。

(3) 可計算性(Computable)：P, Q$\in G_1$，存在一有效率的演算法可計算$e(P,Q)$。

- 雙線性配對計算下的難問題：

1. Discrete Logarithm Problem(DLP)：給定 P, QϵG_1，找到$Q = a \cdot P$中的$a \epsilon Z_q^*$是困難的。
2. Computational Diffie-Hellman Problam (CDHP)：給$(P, aP, bP) \epsilon G_1^3$，$a, b \epsilon Z_q^*$，計算$abP$。
3. Decisional Diffie-Hellman Problem(DDHP)：給$(P, aP, bP, cP) \epsilon G_1^4$，$a, b, c \epsilon Z_q^*$，決定$c = ab \bmod q$是否成立。
4. Gap Diffie-Hellman (GDH) group： 若在G_1中求 DDHP 是簡單但求 CDHP 是困難的，便稱此 group 為 GDH group。

B.基本架構

在我們的免憑證代理簽名系統中有幾個主要的步驟

1. Setup:

   由 KGC 所執行的線性時間演算法，主要目的是產生 KGC 自有的金鑰以及提供給使用者的系統參數。
2. Partial-Secret-Key-Extract:

   同樣是由 KGC 執行的線性時間演算法，以使用者的 ID 為輸入，分別產生不同使用者持有的部分私鑰。
3. User-key-Generation

   使用者得到 KGC 產生的部分私鑰後，會執行幾個步驟以得到自己的金鑰配對。
4. Proxy phase

   原簽章者產生授權給代理簽章者的步驟。
5. Del-Verify

   代理簽章者確認授權是合法的。
6. Proxy-key-Gen

   代理簽章者利用原簽章者的授權產生代理簽章金鑰。
7. Sign

   利用 KGC 產生的系統參數、使用者的私鑰、需要簽章的文件以及代理簽章者的代理簽章金鑰產生一份簽章。
8. Verify

   是一個決定型演算法。得到簽章後，利用 KGC 提供的系統參數、使用者 ID、使用者的公鑰、訊息以及訊息的簽章為輸入，產生的結果為此簽章是否合法。

C. 安全性要求

在免憑證簽章系統中，主要的可能攻擊者有兩種，提出的方法必須可以克服這兩種攻擊者。第一種：KGC 是安全的，攻擊者不知道 KGC 的 master-key，攻擊者會去嘗試竄改使用者的公鑰或者私鑰，想辦法製作出一份可以通過確認者的簽名。

第二種：KGC 是不安全的，攻擊者知道 KGC 的 master-key，攻擊者藉此可以輕易地得到使用者的部分私鑰，但是破壞者沒辦法得知使用者計算出的相對應私鑰。

而因為我們之後會將此簽章系統延伸到盲簽章系統，故我們也要考慮在盲簽章下的安全性要求，而在盲簽章的安全性中，首要考量就是不合法的簽章者可以解開經過盲處理的文件。

三、提案方式（免憑證代理簽名）

我們提出的方法主要包括了三個參與者：一個可信任的第三方 KGC、原有簽章者 A、代理簽章者 B。執行期有以下幾種步驟：KGC 系統設定(Setup)、使用者部分金鑰產生、使用者私有金鑰產生、代理簽章產生、代理簽章認證、代理簽章金鑰產生、文件簽章、簽章認證。總共七個步驟。

1. Setup:

KGC 會執行下列幾個步驟：

- 首先指定出G_1、G_2、q、e、P，如同我們在背景知識的設定。
- 選擇一 $s \in Z_q^*$稱為 KGC 的 master-key，並且設定 KGC 的公鑰P_{pub}=sP
- 選擇兩個 hash functions

  $H1: \{0,1\}^* \rightarrow G_1$

  $H2: \{0,1\}^* \times G_2 \rightarrow Z_q^*$

接著 KGC 會公佈系統參數給使用者

<G_1,G_2, q,e,P, P_{pub},H1,H2>

2. Key-Generation

- Partial-Secret-Key-Extract：

  KGC 利用原簽章者 A 的身分資訊ID_A、及代理簽章者的身分資訊 ID_B，得出 $Q_A = H1(ID_A)$，$Q_B = H1(ID_B)$。

  接著 KGC 會利用剛剛選定的亂數 s，產生使用者分別的部分私鑰。

  $D_A = s \cdot Q_A$，$D_B = s \cdot Q_B$

  分別是 A 和 B 的部分私鑰

- Set-Secret-Number

  簽章者 A、代理簽章者 B 首先會分別選一亂數

  $X_A \in Z_q^*$，$X_B \in Z_q^*$

- Set-Secret-key

  簽章者 A、代理簽章者 B 設定各自的私鑰

  $S_A = X_A \cdot Q_A$，$S_B = X_B \cdot Q_B$

- Set-Public-Key

  簽章者 A、代理簽章者 B 設定各自的公鑰

  $P_{KA} = X_A \cdot P$，$P_{KB} = X_B \cdot P$

3. Proxy-Phase

原始簽章者 A 會給代理簽章者 B 一個代理的授權m_w。

原始簽章者 A 計算下列各值：

$K_A \in Z_q^*$，$r_A = e(P,P)^{K_A}$，$h_A = H2(m_w, r_A)$

$\sigma_{proxy} = h_A \cdot (D_A + S_A) + K_A P$

接著傳送$(m_w, r_A, \sigma_{proxy})$給代理簽章者 B。

4. Del-Verify

代理簽章者 B 得到了$(m_w, r_A, \sigma_{proxy})$後會執行、計算以下步驟，以確認這是由原簽章者 A 所簽發的

首先 B 先利用系統參數計算出：

$h_A = H2(m_w, r_A)$

$Q_A = H1(ID_A)$

接著驗算下列式子是否成立：

$e(\sigma_{proxy}, P) = e(Q_A, P_{KA} + P_{pub})^{h_A} \cdot r_A$

Correctness：

$$e(\sigma_{proxy}, P) = e(h_A \cdot (D_A + S_A) + K_A P, P)$$
$$= e(h_A \cdot (D_A + S_A), P) \cdot e(K_A P, P)$$
$$= e(D_A + S_A, P)^{h_A} \cdot r_A$$
$$= e(s \cdot Q_A + X_A \cdot Q_A, P)^{h_A} \cdot r_A$$
$$= e(Q_A, (s + X_A)P)^{h_A} \cdot r_A$$
$$= e(Q_A, P_{KA} + P_{pub})^{h_A} \cdot r_A$$

5. Proxy-Key-Gen

   代理簽章者 B 產生一代理簽章用的金鑰 d_{proxy}：$h_A \cdot (S_B + D_B) + \sigma_{proxy}$

6. Sign

   使用者 C 想要對一份文件 m 作簽章

   代理簽章者 B 會執行以下步驟來對 m 作出一份代理 A 的簽章

   $K_B \in Z_q^*$，$V_B = K_B \cdot H1(m)$，$U = K_B \cdot P$

   $S = d_{proxy} + V_B$

   最後產生對 m 的簽名為：$\sigma = (r_A, m_w, S, U)$

7. Verify

   驗證下列式子是否成立，可得此簽名是否為合法的簽名：

   $$e(S,P) = (e(Q_A, P_{KA} + P_{pub}) \cdot e(Q_B, P_{KB} + P_{pub}))^{h_A} \cdot r_A \cdot e(H1(m), U)$$

   Correctness

$$e(S,P) = e(d_{proxy} + V_B, P)$$
$$= e(d_{proxy}, P) \cdot e(V_B, P)$$
$$= e(h_A \cdot (S_B + D_B) + \sigma_{proxy}, P) \cdot e(V_B, P)$$
$$= e(S_B + D_B, P)^{h_A} \cdot e(\sigma_{proxy}, P) \cdot e(V_B, P)$$
$$= e(S_B + D_B, P)^{h_A} \cdot e(Q_A, P_{KA} + P_{pub})^{h_A} \cdot r_A \cdot e(V_B, P)$$
$$= e(Q_B, P_{KB} + P_{pub})^{h_A} \cdot e(Q_A, P_{KA} + P_{pub})^{h_A} \cdot r_A \cdot e(K_B \cdot P, P)$$
$$= (e(Q_A, P_{KA} + P_{pub}) \cdot e(Q_B, P_{KB} + P_{pub}))^{h_A} \cdot r_A \cdot e$$

四.安全性分析

在免憑證簽章系統中，破壞者可能企圖去偽造簽章，而如此的破壞者可能會有兩種可能，其一為破壞者知道 KGC 的 master-key s，所以他可以算出使用者的部分私鑰，但是即使如此，在我們的系統中，偽造簽名者做出簽章通過驗證還必須試著去計算出簽章者產生的簽章用私鑰，我們以簽章者 A 為例，由本文章前面章節可以得知，系統中簽章者 A 的私鑰計算為：

$$S_A = X_A \cdot Q_A，X_A \in Z_q^*，Q_A = H1(ID_A) \in G_1$$

根據系統設定，Q_A是公開的，若偽造者知道X_A便可以算出S_A加以偽造簽章者。但偽造者唯一可以得到X_A的機會只有可能從簽章者 A 公佈的公鑰$P_{KA} = X_A \cdot P$中計算出來，但是根據雙線性配對下的 Discrete Logarithm Problem 可以得知若是偽造者知道P_{KA}及P，從中求得X_A是困難的，如此可知，若破壞者想要偽造一份簽名是困難的。

而在我們的系統裡，還必須去考慮偽造代理簽章σ_{proxy}的可能，但從計算式：$\sigma_{proxy} = h_A \cdot (D_A + S_A) + K_A P$中，我們可以發現，若是破壞者想要偽造$\sigma_{proxy}$，前提是他可以偽造出$S_A$，根據前述已知$S_A$是難以取得的，因此我們可知偽造$\sigma_{proxy}$亦是困難的。

另外我們也知道在免憑證簽章系統下，由於公鑰部分沒有發於憑證，於是另一種破壞者很輕易的就可以偽造簽章者公鑰的部分，也會企圖利用改變公鑰的部分去產生一個合法的簽章，但是在我們的系統中，簽章及驗證的步驟都包含了簽章者的私鑰及 KGC 的金鑰，單是改變簽章者的公鑰部分，並沒有辦法去偽造出簽名。

由此可知我們的系統可以抵擋免憑證簽章系統裡面的這兩種攻擊者。

五.提案方式擴張至盲簽名

我們提出的方法不只適用於一般的簽名方

案，更可以提升至適用於盲簽名的部分。要做到這點，只需要在原本的架構下改變一些作法。

在原方案的第 6 步驟，首先使用者先將要加簽的訊息m作盲化處理： $m' = t \cdot H1(m)$，$t \epsilon Z_q^*$

m'即為盲化後的文件，也就是將傳輸給簽章者加簽的文件。

簽章者原本的 sign 過程也必須稍做調整：

$K_B \in Z_q^*$，$V_B = K_B \cdot m'$，$U = K_B \cdot P$

$S = d_{proxy} + V_B$

而認證者也跟著做了調整 $e(S,P) = e(Q_A, P_{KA} + P_{pub}) \cdot e(Q_A, P_{KA} + P_{pub})^{h_A} \cdot r_A \cdot e(H1(m), U)^t$

Correctness:

$$e(S,P) = e(d_{proxy} + V_B, P)$$
$$= e(d_{proxy}, P) \cdot e(V_B, P)$$
$$= e(h_A \cdot (S_B + D_B) + \sigma_{proxy}, P) \cdot e(V_B, P)$$
$$= e(S_B + D_B, P)^{h_A} \cdot e(\sigma_{proxy}, P) \cdot e(V_B, P)$$
$$= e(S_B + D_B, P)^{h_A} \cdot e(Q_A, P_{KA} + P_{pub})^{h_A} \cdot r_A \cdot e(V_B, P)$$
$$= e(Q_B, P_{KB} + P_{pub})^{h_A} \cdot e(Q_A, P_{KA} + P_{pub})^{h_A} \cdot r_A \cdot e(K_B \cdot m', P)$$
$$= e(Q_B, P_{KB} + P_{pub})^{h_A} \cdot e(Q_A, P_{KA} + P_{pub})^{h_A} \cdot r_A \cdot e(K_B \cdot t \cdot H1(m), P)$$
$$= (e(Q_A, P_{KA} + P_{pub}) \cdot e(Q_B, P_{KB} + P_{pub}))^{h_A} \cdot r_A \cdot e(H1(m), U)^t$$

根據以上作法便可以使我們的系統更進一步的運用於盲簽章上。

六.盲簽名的安全性分析

在盲簽名中可能遇到的危險為不合法的簽章者可以解開經過盲處理的文件。在我們的方法中，將文件盲處理的方式為：

取一亂數 $t \epsilon Z_q^*$，$m' = t \cdot H1(m)$。

而$H1: \{0,1\}^* \rightarrow G_1$，可以得知根據雙線性配對下的 Discrete Logarithm Problem，若給破壞者$H1(m)$及m'，從中求 t 是困難的，所以不合法的簽章者要從中取得真正的 m 是困難的。

七.結論

免憑證密碼系統因為不需驗證金鑰憑證，節省了許多複雜的計算，預計將可廣泛的應用於計算能力有限的電子機器像是手機或是 PDA 上。在這篇文章中，我們介紹了一個基於免憑證簽章下的代理簽章系統，除了代理簽章者以外，沒有人可以做簽章的工作，我們同時也證實了這套系統的安全性。在此之前雖然也有人提出了此種架構的系統，但是我們的更將此種系統提升至可應用於盲簽章下，在盲簽章的架構下可以使得我們的系統得到更廣泛的應用。

八.參考文獻

[1] S. Al-Riyami, K. Paterson, “Certificateless public key cryptography”, Advances in Cryptology-Asiacrypt’03, Springer-Verlag, LNCS 2894, 2003, pp.452-473.

[2] D Chaum, “Blind signatures for Untraceable Payments”, Advances in Cryptology-Crypto’82, Plenum Press, 1983, pp:199-203.

[3] Z. Dong, H. Zheng, K. Chen and W. Kou. “ID-based proxy blind signature”, Proceedings of the 18th International Conference on Advanced Information Networking and Application (AINA’04).

[4] K. G. Paterson, “Id-based signatures from pairings on elliptic curves”, Cryptology ePrint Archive, 2002, Report 2002/004,.

[5] S. Hwang and C. Chen, “A new multi proxy

signature scheme", In proc. IWCNS 2000,pp. 134-138

[6] X. Hong and K. Chen, "Secure key-Insulated proxy signature scheme for mobile agent"

[7] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures: Delegation of the power to sign messages", IEICE Trans., 1996, E79-A, (9), pp. 1338-1354.

[8] N. P. Smart, "An identity based authenticated key agreement protocol based on the weil pairing", Electronic Letters, 2002, 630-632.

[9] A. Shamir, "Identity based cryptosystems and signature", In Proc. Crypto 84, LNCS-196, Springer Verlag, 1985, pp. 47-53

[10] Z. Shao, "Proxy signature scheme based on factoring", Information Processing Letter Vol 85(3), 2003, pp. 137-143

[11] B. Zhan and Q.Xu "Certificateless Proxy Blind Signature Scheme from Bilinear Pairings"